

대한민국 특허청
KOREAN INTELLECTUAL
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원번호 : 10-2002-0063495
Application Number PATENT-2002-0063495

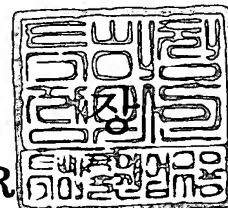
출원년월일 : 2002년 10월 17일
Date of Application OCT 17, 2002

출원인 : 한국전자통신연구원
Applicant(s) Electronics and Telecommunications Research Institute



2003 년 01 월 17 일

특 허 청
COMMISSIONER





【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0001
【제출일자】	2002. 10. 17
【발명의 명칭】	스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 방법
【발명의 영문명칭】	THE REORGANIZATION METHOD IN THE RAID STRIPING SYSTEM
【출원인】	
【명칭】	한국전자통신연구원
【출원인코드】	3-1998-007763-8
【대리인】	
【성명】	장성구
【대리인코드】	9-1998-000514-8
【포괄위임등록번호】	2001-038646-2
【대리인】	
【성명】	김원준
【대리인코드】	9-1998-000104-8
【포괄위임등록번호】	2001-038648-7
【발명자】	
【성명의 국문표기】	박유현
【성명의 영문표기】	BAK, Yuhyeon
【주민등록번호】	731123-1024522
【우편번호】	302-766
【주소】	대전광역시 서구 탄방동 산호아파트 102-409
【국적】	KR
【발명자】	
【성명의 국문표기】	김창수
【성명의 영문표기】	KIM, Chang-soo
【주민등록번호】	691206-1533317
【우편번호】	305-762
【주소】	대전광역시 유성구 전민동 464-1 엑스포아파트 410-704
【국적】	KR

【발명자】**【성명의 국문표기】**

강동재

【성명의 영문표기】

KANG,Dong Jae

【주민등록번호】

721213-1067419

【우편번호】

302-723

【주소】

대전광역시 서구 관저동 1144 구봉마을아파트 708-103

【국적】

KR

【발명자】**【성명의 국문표기】**

김영호

【성명의 영문표기】

KIM,Young Ho

【주민등록번호】

731210-1382411

【우편번호】

305-350

【주소】

대전광역시 유성구 가정동 236-1 (16/5) 신관-118

【국적】

KR

【심사청구】

청구

【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인
 장성구 (인) 대리인
 김원준 (인)

【수수료】**【기본출원료】**

15 면 29,000 원

【가산출원료】

0 면 0 원

【우선권주장료】

0 건 0 원

【심사청구료】

7 항 333,000 원

【합계】

362,000 원

【감면사유】

정부출연연구기관

【감면후 수수료】

181,000 원

【기술이전】**【기술양도】**

희망

【실시권 허여】

희망

【기술지도】

희망

【첨부서류】

1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

스트라이핑(Striping) 기법을 이용한 RAID(Redundant Array of Inexpensive(or Independent) Disks) 시스템에서의 데이터 재구성 방법을 개시한다.

본 발명은 스트라이핑 기법을 이용하여 데이터를 저장하는 시스템에서 디스크가 추가되는 경우, 데이터를 즉시 재구성하지 않고 쓰기 연산이 발생한 블록에 대해서만 재구성 과정을 수행하는 것을 특징으로 한다.

본 발명에서는, 디스크를 추가한 후 데이터 쓰기 연산이 요청되면 재구성된 후의 위치를 계산하여 그 위치에 데이터를 기록한다. 만일, 기록하고자 하는 위치에 다른 데이터가 존재한다면 그 데이터에 대해서도 재구성된 후의 위치를 계산하여 그 위치에 데이터를 기록한다. 최악의 경우, 요청된 블록에 대해 연쇄적으로 발생하는 재구성 블록의 수가 커지는 경우도 발생할 수 있으나, 이러한 연산 과정을 통해 일반 디스크 I/O에 대한 요청을 금지시키는 시간이 매우 짧아지며, 요청 빈도수가 높은 데이터에 대해 미리 재구성을 수행할 수 있는 효과를 가진다.

【대표도】

도 5

【명세서】**【발명의 명칭】**

스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 방법{THE REORGANIZATION METHOD IN THE RAID STRIPING SYSTEM}

【도면의 간단한 설명】

도 1은 전형적인 레이드 시스템에서의 데이터 재구성 전의 디스크 상태를 예시적으로 도시한 도면,

도 2는 전형적인 레이드 시스템에서의 데이터 재구성 후의 디스크 상태를 예시적으로 도시한 도면,

도 3은 본 발명에 따른 방법에 의해 쓰기 연산이 요청된 블록에 대해서 재구성한 후의 디스크 상태를 예시적으로 도시한 도면,

도 4는 도 3의 블록 재구성시 변화되는 재구성 테이블 상태 도면,

도 5 및 도 6은 본 발명의 바람직한 실시예에 따른 스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 과정의 흐름도,

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<6> 본 발명은 RAID(Redundant Array of Inexpensive(or Independent) Disks : 이하 레이드라 함) 시스템에서의 데이터 재구성 처리 기법에 관한 것으로, 특히, 디스크의 입출

력 성능 향상을 위하여 스트라이핑(Striping)으로 데이터를 저장하는 레이드 시스템에서의 데이터 재구성 방법에 관한 것이다.

- <7> 인터넷 사용의 대중화는 작업 환경의 급속한 변화와 함께 저장되어야 할 데이터 용량의 폭발적인 증가를 가져왔다. 더욱이, 멀티미디어의 발전으로 인해 일반 컴퓨터에서 처리해야 할 데이터 크기가 매우 커졌으며, 컴퓨터 사용자의 증가로 인해 처리해야 할 데이터 용량 또한 증가하였다.
- <8> 그러나, 하나의 서버에 접속되어 데이터를 저장하고 관리하는 클라이언트 서버의 데이터 관리 시스템이나 파일서버를 기반으로 하는 네트워크 파일 시스템과 같은 일반 자료 저장 시스템들은 급격하게 증가된 데이터를 처리하기에는 한계가 있었다.
- <9> 이렇게 증가된 데이터들을 효율적으로 처리하기 위하여 최근에는 자료 저장 시스템에서 레이드 기술을 채택하고 있다. 레이드의 기본적인 개념은 작은 드라이브들을 연결하여 큰 드라이브 하나를 대체함으로써 이들이 갖는 모든 입출력 대역폭을 동시에 사용하여 최대 데이터 전송 대역폭을 극대화하는 것이다.
- <10> 레이드 기술은 그 특성에 따라 여러 단계를 제공하는데, 레이드 레벨 0과 3, 4, 5는 디스크 배열을 구성하는 장치들에 데이터를 분산하여 저장하는 방법이다. 즉, 모든 데이터를 스트라이핑 단위(Striping unit)로 각 디스크에 순차적으로 써 나가는 스트라이핑 기법으로 디스크로의 입/출력을 동시에 수행하기 때문에 병렬 특성이 높아진다.
- <11> 한편, 스트라이핑 기법으로 데이터를 저장하는 시스템에서 새로운 디스크를 추가하여 용량을 늘리고자 하는 경우, 일련의 데이터 재구성 과정을 수행한다.

- <12> 도 1 및 도 2는 이러한 재구성 과정을 설명하기 위한 예시 도면으로서, 도 1은 추가되기 전의 디스크 상태, 도 2는 디스크 추가 후 재구성한 상태를 나타낸다.
- <13> 하지만, 이러한 기존의 재구성 방법은 재구성을 위해 시스템의 수행을 중단시킨 후 전체 디스크의 내용을 읽어서 배치 방식에 따라 다시 디스크에 저장하여야 하며, 연산 중에 일반적인 디스크 I/O 연산을 수행할 수 없다는 단점이 있다.
- <14> 따라서, 데이터 재구성을 위한 디스크 연산 수행 시간이 필요 이상으로 소요되어 시스템 성능에 있어서 오버헤드가 발생된다는 문제가 있었다.

【발명이 이루고자 하는 기술적 과제】

- <15> 본 발명은 상술한 종래 기술의 문제점을 해결하기 위하여 안출된 것으로, 스트라이핑을 하는 시스템에서 용량 확장을 위한 디스크를 추가할 때 기존에 저장되어 있는 데이터 전체를 재구성하지 않고 사용자의 요청 중에서 쓰기 연산이 발생한 데이터에 대해서만 재구성을 수행하여 디스크 추가 후 곧바로 서비스를 진행할 수 있도록 한 스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 방법을 제공하는데 그 목적이 있다.
- <16> 이러한 목적을 달성하기 위한 본 발명의 일 실시예에 따르면, 스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 방법에 있어서, 시스템의 저장 매체에 추가 디스크가 존재하는지를 판단하는 제 1 단계와; 제 1 단계의 판단 결과, 추가 디스크가 존재하는 경우, 디스크 쓰기 연산에 대해서 재구성 요청 위치에 임의의 데이터가 저장되어 있는지를 판단하는 제 2 단계와; 제 2 단계의 판단 결과, 재구성 요청 위치에 임의의 데이터가 저장되어 있지 않는 경우, 해당 위치로 재구성 요청 데이터를 이동시키는 제

3 단계를 포함하는 것을 특징으로 하는 스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 방법을 제공한다.

【발명의 구성 및 작용】

- <17> 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예에 대하여 상세하게 설명한다.
- <18> 도 3은 본 발명에 따른 방법에 의해 쓰기 연산이 요청된 블록에 대해서 재구성한 후의 디스크 상태를 예시적으로 도시한 도면이며, 도 5는 이러한 블록 재구성 방법의 흐름도를 나타낸다.
- <19> 먼저, 단계(S500)에서는 임의의 논리주소(A)에 대한 I/O가 요청되는지를 판단하고, I/O가 요청되면 단계(S502)로 진행하여 재구성 테이블에 저장된 물리주소(A')를 읽는다.
- <20> 단계(S504)에서는 읽혀진 물리주소(A')가 초기값인지를 판단한다. 물리주소(A')가 초기값이 아니라면 해당 블록을 물리주소(A')로 I/O하도록 하나(S506), 물리주소(A')가 초기값이라면, 단계(S508)로 진행하여 해당 블록이 읽기 연산인지 쓰기 연산인지를 판단한다.
- <21> 즉, 본 발명에서는 도 1의 초기 디스크 상태에서 디스크 1개가 추가되는 경우, 사용자로부터 쓰기 요청된 블록에 대해서만 재구성을 수행하는 것을 특징으로 한다. 예를 들어, 도 1의 초기 디스크 상태에서 사용자로부터 논리주소 12번에 대해 쓰기 연산이 요청되면 12번 블록에 대한 재구성 후의 위치는 다음과 같이 계산된다.
- <22> $12 \% 4 = 0$ (나머지)
- <23> $12 / 4 = 3$ (몫)

- <24> 즉, 도 3에 도시한 바와 같이 12번 블록에 대한 재구성 후의 위치는 0번 디스크의 3번째 블록에 위치하여야 한다(디스크 순서는 좌로부터 0, 1, 2, 3..., 블록 순서는 좌측 상단부터 지그재그 형식으로 0, 1, 2, 3...의 순서이다).
- <25> 단계(S508)의 판단 결과, 쓰기 연산으로 판단되는 경우(S510), 단계(S511)로 진행하여 논리주소(A)의 재구성 물리주소(A')를 계산하고, 계산된 재구성 물리주소(A')가 추가 디스크에 위치하는지를 판단한다(S512).
- <26> 단계(S512)의 판단 결과, A번 블록에 대한 재구성 물리주소(A')의 디스크 번호가 새로 추가된 디스크 번호라면, 단계(S513)로 진행하여 논리주소(A)의 내용을 새로운 위치에 저장한다.
- <27> 또한, 단계(S514)에서와 같이, 재구성 테이블 상의 A의 물리주소 위치에 F(재구성 수행하지 않았음)로 표시되어 있는 내용을 실제 물리주소인 A'으로 갱신한다. 이러한 재구성 테이블의 논리주소 A의 내용을 갱신할 때는 다중 호스트 서비스를 위해 잠금 기능을 사용한다.
- <28> 즉, 본 발명은, 재구성 테이블을 갱신할 때에는 항상 잠금 기능을 설정 및 수행하며, 갱신이 끝나면 잠금 기능을 즉시 해제시키는 것을 특징으로 한다. 논리블록 3은 이러한 과정을 거친다.
- <29> 만일, 단계(S512)의 판단 결과, 요청된 A번 블록(예컨대, 12번 블록)의 재구성 물리주소(0번 디스크의 3번째 블록)의 디스크 번호가 기존의 디스크라면, 새로 결정된 물리주소에는 이미 다른 데이터가 저장되어 있는 바, 이미 그 위치(A')에 저장되어 있는 블록(9번 블록)을 먼저 이동시켜야 한다.

- <30> 이러한 과정은 연쇄적으로 발생하며, 이 과정 중에 시스템을 재동작시킬 경우도 있기 때문에 요청된 A번 블록의 내용은 연쇄 재구성 과정이 끝날 때까지 디스크에 안전하게 저장되어야 한다.
- <31> 따라서, 단계(S515)에서와 같이, A번 블록의 내용을 새로 추가된 디스크의 하위 주소 부분(A'')에 저장한다.
- <32> 이때, 본 발명에서는, 요청된 쓰기 블록에 대해 추가된 디스크의 끝부분부터 저장하기 위해서 자유공간관리자가 그 영역을 관리하는 것을 특징으로 한다.
- <33> 한편, A번 블록이 안전하게 저장되면, 재구성 테이블의 A번 블록에 대해 잠금을 획득하고 물리주소를 A''으로 갱신한 후에 잠금을 해지한다(S516).
- <34> A번 블록(12번 블록)의 연산이 끝난 후에, A로 인한 연쇄적 재구성을 수행해야 하는 블록(9, 7)들을 계속 계산하여 그 재구성 위치가 새로 추가된 디스크로 되도록 한다. 본 실시예에서는 요청된 블록이 12번 블록이기 때문에 7번 블록(B)이 최종적으로 새로운 디스크에 저장되게 된다.
- <35> 따라서, 결정된 최종 블록(B번 블록)의 재구성 위치(B' : 3번 디스크의 1번 물리블록)를 계산하고(S517), 그 위치에 B번 블록(7번 블록)의 내용을 이동시킨다(S518).
- <36> 마찬가지로, 재구성 테이블의 7번 블록의 위치를 잠금을 통해서 갱신한다(S519).
- <37> 최종 연쇄블록에 대한 재구성이 끝나면 차례로 따라 올라가면서 각 블록에 대해서 재구성 연산을 수행하면 된다. 이러한 과정은 대상 블록이 요청된 블록(A)이 될 때까지 진행된다.

- <38> 이 과정을 살펴보면, 7번 블록이 원래 있던 위치(1번 디스크의 2번 물리블록)에 재구성 위치가 결정되는 논리블록을 계산하면 9번 블록이 된다(S520).
- <39> 따라서, 9번 블록의 내용을 1번 디스크의 2번 물리블록으로 이동시키고(S522), 마찬가지로 재구성 테이블상의 9번 블록의 위치를 잠금을 통해서 갱신한다(S523).
- <40> 9번 블록이 원래 있던 위치(0번 디스크의 3번 물리블록)로 이동할 논리블록 번호는 12번이다. 이 논리블록은 사용자로부터 요청된 블록번호이고, 이 내용은 새로 추가된 디스크의 끝부분 영역에 저장되어 있고, 그의 물리주소는 A''으로 되어 있다.
- <41> 따라서, A''에 저장되어 있는 물리블록 A(12번 블록)의 내용을 0번 디스크의 3번 물리블록으로 이동시키고, 재구성 테이블상의 물리주소를 A''에서 (0, 3)으로 갱신한다(S525)(S526).
- <42> 한편, 재구성 연산이 수행된 블록들의 이동 위치는 재구성 테이블에 저장되며, 재구성이 된 블록인지의 여부는 재구성 테이블 상의 값이 초기화 상태인지 여부를 확인함으로써 알 수 있다.
- <43> 만일, 초기 상태가 아니라면 재구성이 수행된 블록이다. 따라서, 쓰기 연산이 요청되었을 경우 재구성이 되어 있는 블록인지를 먼저 확인한 후 재구성되어 있지 않은 블록이라면 연관 블록들을 바른 위치로 이동시키는 추가적인 연산이 필요하며, 재구성이 되어 있는 블록이라면 재구성 테이블에 저장된 위치 정보를 이용하여 디스크 I/O를 수행한다.

- <44> 읽기 연산 또한 재구성 테이블을 참조하여 재구성 테이블이 초기화되어 있다면 원래의 위치를, 그렇지 않다면 재구성 테이블 상의 위치 정보를 이용하여 디스크 I/O를 수행한다.
- <45> 논리주소 12번에 대한 블록 재구성이 끝난 후의 재구성 테이블 값은 도 4에 도시한 바와 같다. 도 4에 도시한 바와 같이, ()안의 숫자 쌍 중에서 앞의 숫자는 디스크 번호, 뒤 숫자는 물리주소 번호이며, F는 초기 상태를 나타낸다.
- <46> 모든 재구성 테이블의 변경에는 잠금을 설정함으로써 다중 사용자로부터의 접근을 허용할 수 있다.
- <47> 이러한 재구성 과정이 종료되면 도 4의 재구성 테이블을 초기화한다. 이러한 초기화 과정은 디스크가 추가되어 부분 재구성이 시작되는 시점에서 수행할 수도 있다.

【발명의 효과】

- <48> 앞서 상세히 설명한 바와 같이 본 발명은, 스트라이핑을 하는 시스템에서 디스크를 추가할 때 종래와 같이 전체 디스크의 데이터를 재구성하지 않고 사용자에게 의해 쓰기 연산이 요청된 블록에 대해서만 데이터를 재구성함으로써, 별도의 재구성 연산에 의해 시스템을 정지해야 하는 시간을 줄일 수 있다.
- <49> 이상, 본 발명을 실시예에 근거하여 구체적으로 설명하였지만, 본 발명은 이러한 실시예에 한정되는 것이 아니라, 후술하는 특허청구범위의 요지를 벗어나지 않는 범위 내에서 여러 가지 변형이 가능한 것은 물론이다.
- <50> 성능적인 측면을 고려하여, 청구범위에서는 블록번호가 적은 순서로 데이터를 옮기도록 한다.

【특허청구범위】**【청구항 1】**

스트라이핑(Striping) 기법을 이용한 레이드(RAID) 시스템에서의 데이터 재구성 방법에 있어서,

상기 시스템의 저장 매체에 추가 디스크가 존재하는지를 판단하는 제 1 단계와;

상기 제 1 단계의 판단 결과, 추가 디스크가 존재하는 경우, 디스크 쓰기 연산에 대해서 재구성 요청 위치에 임의의 데이터가 저장되어 있는지를 판단하는 제 2 단계와;

상기 제 2 단계의 판단 결과, 상기 재구성 요청 위치에 임의의 데이터가 저장되어 있지 않는 경우, 해당 위치로 재구성 요청 데이터를 이동시키는 제 3 단계를 포함하는 것을 특징으로 하는 스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 방법 .

【청구항 2】

제 1 항에 있어서,

상기 방법은,

상기 제 2 단계의 판단 결과, 상기 재구성 요청 위치에 임의의 데이터가 저장되어 있는 경우, 요청된 디스크 쓰기 연산 블록을 상기 추가 디스크의 끝부분부터 순차 저장 하되 상기 저장되는 물리적 정보를 재구성 테이블의 요청된 블록 정보에 갱신하는 단계를 더 포함하는 것을 특징으로 하는 스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 방법 .

【청구항 3】

제 1 항 또는 제 2 항에 있어서,

상기 방법은,

상기 갱신 단계 수행 후, 요청된 블록으로 인한 연쇄 쓰기 블록들을 반복적으로 계산한 다음, 연쇄 재구성이 끝나는 블록을 탐색하는 단계와;

탐색된 블록을 상기 추가 디스크로 이동시키고 그에 따른 물리적 정보를 상기 재구성 테이블에 반영하는 단계와;

상기 추가 디스크에 저장된 블록의 원래 자리로 이동시킬 블록을 계산하여 해당 블록을 이동시킨 다음, 상기 재구성 테이블에 반영하는 단계로 이루어지는 것을 특징으로 하는 스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 방법.

【청구항 4】

제 3 항에 있어서,

상기 블록 이동 연산 과정은 요청된 쓰기 연산 블록의 재구성이 종료될 때까지 수행되는 것을 특징으로 하는 스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 방법.

【청구항 5】

제 1 항에 있어서,

상기 재구성 테이블을 수정하는 경우에는 잠금(lock) 기능을 사용함으로써 동시 사용자 서비스를 제공하는 것을 특징으로 하는 스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 방법.

【청구항 6】

제 1 항에 있어서,

상기 방법은,

디스크에 저장된 모든 데이터에 대하여 상기 제 3 단계의 수행 후 또는 디스크 추가 연산이 발생한 초기에 상기 재구성 테이블을 초기화하는 것을 특징으로 하는 스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 방법.

【청구항 7】

제 2 항에 있어서,

상기 방법은,

요청된 쓰기 블록에 대해 추가된 디스크의 끝 부분부터 저장하기 위해 자유공간관리자가 그 영역을 관리하는 것을 특징으로 하는 스트라이핑 기법을 이용한 레이드 시스템에서의 데이터 재구성 방법.

【도면】

【도 1】

0	3	1	4	2	5		
6	9	7	10	8	11		

【도 2】

0	4	1	5	2	6	3	7
8	12	9		10		11	

【도 3】

0	3	1	4	2	5		7
6	12	9	10	8	11		

【도 4】

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
F	F	F	F	F	F	F	(3,1)	F	(1,2)	F	F	(0,3)	F	F

【도 5】

